

CoachGAN: Fast Adversarial Transfer Learning between Differently Shaped Entities

Mehdi Mounsif¹^a, Sébastien Lengagne¹^b, Benoit Thuilot¹ and Lounis Aduane²^c

¹Université Clermont Auvergne, CNRS, SIGMA Clermont, Institut Pascal, F-63000 Clermont-Ferrand, France

²Université de Technologie de Compiègne, CNRS, Heudiasyc, F-60200 Compiègne, France

Keywords: Transfer Learning, Generative Adversarial Networks, Control, Differentiable Models.

Abstract: In the last decade, robots have been taking an increasingly important place in our societies, and shall the current trend keep the same dynamic, their presence and activities will likely become ubiquitous. As robots will certainly be produced by various industrial actors, it is reasonable to assume that a very diverse robot population will be used by mankind for a broad panel of tasks. As such, it appears probable that robots with a distinct morphology will be required to perform the same task. As an important part of these tasks requires learning-based control and given the millions of interactions steps needed by these approaches to create a single agent, it appears highly desirable to be able to transfer skills from one agent to another despite a potentially different kinematic structure. Correspondingly, this paper introduces a new method, CoachGAN, based on an adversarial framework that allows fast transfer of capacities between a teacher and a student agent. The CoachGAN approach aims at embedding the teacher's way of solving the task within a critic network. Enhanced with the intermediate state variable (ISV) that translates a student state in its teacher equivalent, the critic is then able to guide the student policy in a supervised way in a fraction of the initial training time and without the student having any interaction with the target domain. To demonstrate the flexibility of this approach, CoachGAN is evaluated over a custom tennis task, using various ways to define the intermediate state variables.


1 INTRODUCTION


The development of human civilization was supported by various human features, both in terms of morphology and behavior (Boyden, 2004). Among these evolution pillars, knowledge transfer was a crucial asset as it enabled the specie to quickly embed a wide range of skills and abilities in its descendants. For intellectual tasks as well as physical activities, transfer knowledge can greatly decrease the time to reach expertise.


Reproducing this knowledge transfer within a population of robots is a highly desirable goal but is not free of challenges. Despite recent impressive Reinforcement Learning (RL) advances, current RL methods mostly focus on a single agent with a single assignment and still feature a very low sample efficiency, requiring millions of interaction steps and carefully tuned reward functions to develop suitable

policies. As this process should be repeated for every distinct robot, it is straightforward to understand the relevance of transfer learning across all robotic applications.

Consequently, this work introduces CoachGAN, a new method for the fast transfer of skills between differently shaped agents. Building on the concept of expert/teacher/student where a teacher can use observations made during the expert class to guide its students although the teacher was not the initial transfer target, CoachGAN relies on a two-step adversarial process to train a student agent by using expert knowledge via a common teacher critic. Specifically, the main idea is to repurpose a GAN discriminator (the teacher) trained on a set of expert trajectories to provide an error signal for a student trying to accomplish the same task. As the very essence of this work is to focus on differently shaped entities, it is crucial that the student and the teacher can share a common understanding for the student to use the discriminator evaluation to backpropagate accordingly its error. As such, this paper provides an innovative yet simple way to translate the discriminator's value into a learn-

^a <https://orcid.org/0000-0002-2763-3890>

^b <https://orcid.org/0000-0002-1831-1072>

^c <https://orcid.org/0000-0002-5686-5279>

ing signal *via* intermediate State Variables (ISV). The relevance of this technique is then assessed over a custom Tennis environment, using three different ways to translate the discriminator signal

2 RELATED WORKS

Recently, the transfer of knowledge and skills in deep learning has been a ubiquitous topic. While this domain has generated much interest from its very first moments in learning-based CV (Computer Vision) (Simonyan and Zisserman, 2014; He et al., 2015), it is now hardly expendable and has been applied to an even greater extent in NLP (Natural Language Processing). As a matter of fact, given the impressive quantity of resources required to train the most recent models (Devlin et al., 2018; Martínez-González et al., 2018), it is now very common to use pre-trained weights to initialize the model and fine-tune its parameters on the target task.

The omnipresence of transfer within the classic CV and NLP pools of tasks has nevertheless not particularly influenced the approaches with robotic control tasks where it is more common to rely on Reinforcement Learning (RL). RL has made impressive progress in recent years, from algorithmic advances (Schulman et al., 2017; Haarnoja et al., 2018), to both simulated (Silver et al., 2015; Cruz et al., 2016; Jeong and Lee, 2016) and real-world results (OpenAI et al., 2019). In the RL paradigm, it is more common to train an agent on a given task and modify it afterward to see whether the agent’s previously learned representations can be repurposed (Trapit et al., 2017; Baker et al., 2019). Nonetheless, the fact that architectures are usually shallow (for most classical control settings, including Mujoco’s physics-based environments, 3 layers are usually enough (Schulman et al., 2017)) does not encourage knowledge segmentation through a dimensional bottleneck, like in usual CV/NLP architectures. To broaden an agent’s usability, some works propose entropy-based loss functions (Eysenbach et al., 2018) to increase the agent’s curiosity and exploration as well as meta-Learning methods (Zintgraf et al., 2018). Although these works do enhance the relative reusability of an agent, there exist numerous real-world cases that would rather benefit from transferring knowledge from one entity to a distinct one. In (Mounsif et al., 2019), the authors propose to create a task model, independent from the agent’s body, thus allowing them to transfer it from one agent to another. Although this technique is theoretically powerful, it is in practice limited by the manually-defined interface between the task model

and the agents, heavily impacting its flexibility. This topic is however seldom addressed in recent works. In this view, the CoachGAN approach, through its differentiable chain, can cope with a broad scope of situations and is not limited by the task model expressiveness.

3 METHOD

The CoachGAN approach relies on an adversarial framework to transfer task-specific skills from one expert agent to a student agent with a different morphology.

3.1 GAN Background

Generative Adversarial Networks (GANs), introduced in (Goodfellow et al., 2014), is a powerful concept that aims at estimating a generative model via an adversarial process. In this framework, two networks, the generator, and the discriminator are trained simultaneously to optimize a loss function that involves outperforming the other network. Specifically, the generator’s goal is to output datapoint that is as close as possible to a target distribution while the discriminator loss depends on how much it succeeds at distinguishing the generator output from the real data distribution. Formally, the competition between the generator G and the discriminator D is the minimax objective:

$$\min_G \max_D \mathbb{E}_{x \sim \mathbb{P}_r} [\log(D(x))] + \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [\log(1 - D(\tilde{x}))] \quad (1)$$

where \mathbb{E} represents the averaging operator, \mathbb{P}_r is the real data distribution, \mathbb{P}_g is the model distribution, defined by $\tilde{x} = G(z)$, where z the generator input is sampled from a noise distribution (generally uniform or normal distribution).

3.2 CoachGAN

As opposed to most GANs applications where the generator is the only desired model and where the discriminator is generally discarded, the CoachGAN method uses the discriminator to permit the transfer to the student. The main concept of this approach is to define an intermediate state variable (ISV) that can be evaluated by the (teacher) discriminator, based on representations learned on the expert trajectories, and to which the student can relate. It will then be possible to backpropagate the student error within its model and optimize it based on the teacher discriminator evaluation. The ISV is critical to the Coach-

GAN approach. Indeed, as the expert and the student have different morphologies, it is not straightforward how to directly compare their states/actions. The ISV role is to bridge the gap between the two agents by being the representation of a common, user-defined, intermediate state.

It would have been possible to guide the student by using the distance between the student and the expert’s ISV as a loss value. However, while theoretically viable, this option is practically more complex to set up due to the discriminator pre-convergence. Specifically, in adversarial frameworks, the discriminator often trains faster than the generator, which, in extreme cases, can even prevent the generator from learning as all its samples are rejected by the discriminator. This constitutes an important drawback in most use cases and may require extensive hyperparameter tuning. For this reason, it is more straightforward to rely on the teacher discriminator to train the student generator.

The first stage aims at training the teacher discriminator to create a relevant task critic. Using Equation 1, the objective becomes:

$$\min_{G_T} \max_{D_T} \mathbb{E}_{x_{\text{ISV}} \sim \mathbb{P}_T} [\log(D_T(x_{\text{ISV}}))] + \mathbb{E}_{\tilde{x}_{\text{ISV}} \sim \mathbb{P}_g} [\log(1 - D_T(\tilde{x}_{\text{ISV}}))] \quad (2)$$

where G_T, D_T are respectively the fake expert generator and teacher discriminator, $x_{\text{ISV}} \sim \mathbb{P}_T$ represents ISV vector from the expert dataset and $\tilde{x}_{\text{ISV}} = G_T(z)$ are synthetic ISV vectors from the fake expert generator, which sole purpose is to train the teacher discriminator. Once trained on the expert trajectories, the discriminator, when presented with an ISV, will return a scalar value, ranging from 0 to 1, indicating how likely it is that this sample was generated by the expert.

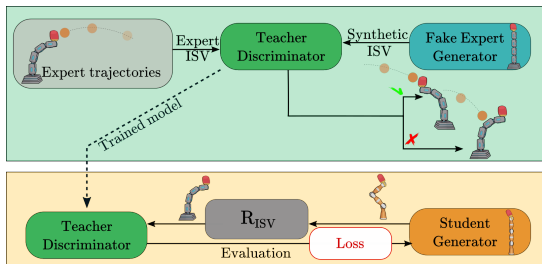


Figure 1: The CoachGAN principle. The green frame depicts the teacher discriminator training process that is then used for training the student generator (orange frame).

Then, for the transfer, a new generator relevant to the student control dimensionality is created and paired with the teacher discriminator. In this configuration, the student loss is:

$$L_S = \|D_T(R_{\text{ISV}}(G_S(z))) - 1\|^2 \quad (3)$$

Equation 3 states that the student generator aims at maximizing the discriminator evaluation given to a synthetic ISV-translated student generator solution. Practically, for a given input z , the student generator solution $G_S(z)$ is passed to the differentiable operator R_{ISV} that computes a discriminator-understandable ISV vector. Given that every function in this chain is differentiable, it is consequently possible to back-propagate the discriminator evaluation within the student’s parameters for an optimization step. Figure 1 shows a schematic view of the method.

4 EXPERIMENTAL SETUP

4.1 Models

Training in the adversarial framework is notoriously difficult (Salimans et al., 2016b; Radford et al., 2016) due to the multiple failure modes (discriminator pre-convergence, generator mode collapse). Furthermore, as the teacher discriminator is expected to be used for the downstream transfer, it is necessary that the discriminator sample evaluation provides a suitable signal, learning-wise. Taking into account these various constraints, the WGAN-GP (Wasserstein GAN with Gradient Penalty (Salimans et al., 2016a)) appeared as a natural architecture candidate because beyond making the generator training easier, it also formulates the discriminator evaluation in a way that enables it to be repurposed. Moreover, as a task-related vector must be passed to the networks, this work implements a conditional WGAN-GP, yielding the following formulation:

$$L_D = \mathbb{E}_{\tilde{x} \sim \mathbb{P}_g} [D(c, \tilde{x})] - \mathbb{E}_{x \sim \mathbb{P}_r} [D(c, x)] + \lambda P_g \quad (4)$$

where c is the task related vector, λ is the gradient penalty weight and P_g the gradient penalty:

$$P_g = \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}} [(\|\nabla_{\hat{x}} D(c, \hat{x})\| - 1)^2] \quad (5)$$

where \hat{x} is a vector interpolated between x and \tilde{x} with an interpolation factor sampled uniformly between 0 and 1.

As the environment featured in this paper yields low-dimensionality observations, the models considered are quite thin, being composed of two hidden layers of 128 units each and ReLU non-linearities. For the generators, the random noise vector is sampled from a 16 dimensions latent space.

4.2 Intermediate State Variables

To demonstrate the usefulness of the CoachGAN approach, a custom 2D-Tennis environment was created. This environment features the playing agent,

a serial manipulator equipped with a bat, and a ball thrown with an initial velocity, as shown in Figure 3a. The agent’s goal is to position its effector in order to intercept the ball. As mentioned in Section 3, the CoachGAN approach relies on an intermediate state variables for allowing the teacher discriminator evaluation to guide the student. One of the main strength of this approach is that these intermediate state variables can be very diverse as long as the student can relate to them through a differentiable model. As an example, three configurations for this approach are provided:

Actual Kinematics: In this first configuration, the intermediate state variable is the effector position. Specifically, for a given ball configuration, the teacher discriminator is trained to distinguish between suitable effector position (that is, from a task expert) and the synthetic position from the fake expert generator. The student generator will output the target joint angles. To evaluate the student generator proposition, a differentiable forward kinematics (FK) model is used to translate the joint angles to the effector position that can thus be assessed by the discriminator, as shown in Figure 2. The FK model used in this configuration has no trainable parameters and is only used to keep track of the gradients.

Approximated Kinematics: This second composition still relies on the effector position, but this time the FK model is replaced by a neural network trained on the student configuration to compute the effector position given joint angles, also shown in Figure 2.

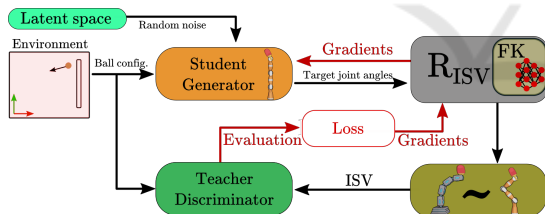


Figure 2: Student Generator (in orange) optimization in experiments **Actual Kinematics** and **Approximated Kinematics**. Discriminator evaluation passes through the R_{ISV} module before reaching the generator.

Ball Rebound Speed: The effector position is a very convenient variable when considering serial manipulators. However, it may be less relevant in cases where agents exhibit a very different structure or in other types of tasks as well. Thus, to illustrate the CoachGAN method applicability, this third configuration demonstrates that it is possible to transfer task knowledge through measurements not directly accessible. Specifically, this case proposes to train the student using the ball velocity. This configuration requires an additional ball speed predictor model, trained to predict ball speed at the next step, given the ball configu-

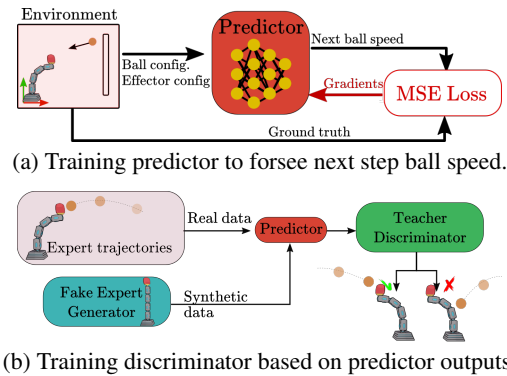


Figure 3: Intermediate training steps for the **Rebound Speed** configuration.

ration and the effector position, as displayed in Figure 3a.

Training the discriminator consequently requires an additional step, during which effector positions, both real and synthetic, are first passed through the predictor. This way, the teacher discriminator learns to classify suitable ball speeds, based on the predictor predictions, see Figure 3b.

Once the teacher discriminator is trained, it is used to train the student generator. As done precedently, the student generator proposed angles result in an effector position, computed either with an analytical or approximated model, which is then used by the predictor to compute the next step ball speed. The discriminator evaluation of this vector is finally used to optimize the student generator’s weights.

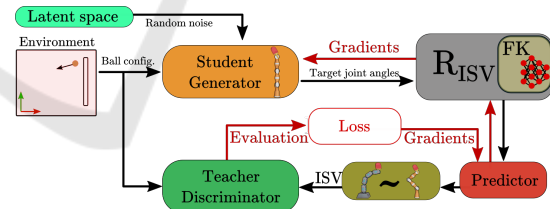


Figure 4: Student generator (in orange) training setup for experiment **Ball Rebound**. Discriminator evaluation goes through the predictor and R_{ISV} before modifying generator parameters.

4.3 Training the Expert

In the experiments presented below, a task expert is required. This expert agent is a 4-DoF serial manipulator, trained using PPO (Schulman et al., 2017), a SOTA RL algorithm. In a MDP (Markov Decision Process) version of the Tennis environment, the agent aims at maximizing the following reward function:

$$r_{\text{tennis}} = \alpha + c \times (\beta + \gamma * \exp(-d)) \quad (6)$$

where α is a small constant that incites the agent to keep the ball over a height threshold for as long as possible. c is the contact flag ranging from 0 to 1 when a contact between the ball and the wall is detected, and goes back to 0 at the next step. β and γ are constant values used to weight the relative importance of accuracy when touching the wall. Finally, d is the vertical offset between the ball and the target on the wall. This reward function broadens the ball impact distribution, thus improving the agent versatility and ultimately providing a more diverse trajectory dataset. Figure 5 shows the mean cumulative reward evolution through training and, in the upper-left corner the impact distribution for a trained agent given various target heights.

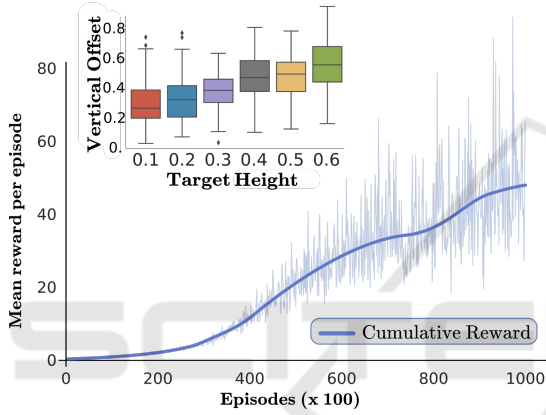


Figure 5: Cumulative reward through training and impact distribution for a set of given height targets.

5 RESULTS

5.1 Effector Position ISV

Once the expert agent is ready, it is used to generate the training dataset. For the two first configurations, that is, **Actual Kinematics** and **Approximated Kinematics**, the proposed ISV is the effector position. As such, each line of the gathered dataset features the initial ball conditions (position and speed) and the expert effector position recorded when contact is detected.

The adversarial pair (teacher discriminator and fake expert generator) is then trained on this trajectory dataset. Figure 6 shows the adversarial losses along training. These alternating curves are usual in adversarial framework and represent one model improving in spite of the other. It is possible to see that the process finally stabilizes. In the upper right corner are displayed fake expert generator ISV solutions for various ball configurations, showing that this model

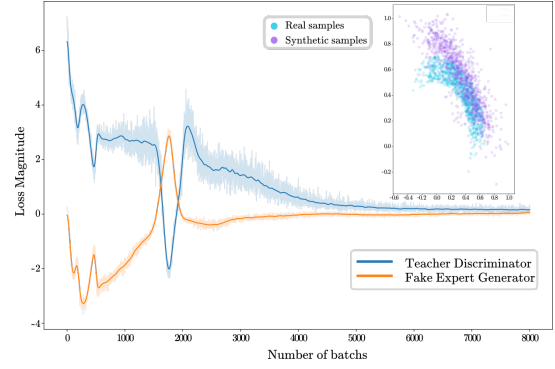


Figure 6: Adversarial losses along training.

decently approximated the real expert behavior.

Figure 7 displays the normalized discriminator evaluation of ISV (effector positions in this case), for a given ball configuration. As can be observed, the discriminator presents a clear preference for ISV along a line following the expected ball path. Furthermore, the discriminator places its highest confidence (strong yellow color) in an area located in the close vicinity of the dataset ground truth (the expert effector position, drawn in green), showing that it has indeed learned from the expert behavior.

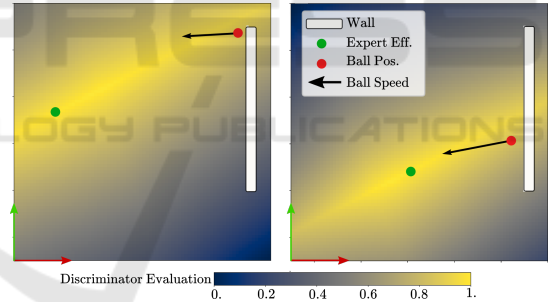


Figure 7: Normalized discriminator evaluation of ISV (effector positions) in the task space for two given starting ball configurations. Higher values (yellow) are favored.

Once trained, the discriminator can be repurposed for training the transfer target, which is the student generator. In this configuration, the task knowledge is transferred towards a 5-DoF serial manipulator. As such, the generator output vector is interpreted as the target joint angles for the student. Then, it is necessary to translate the student generator output into the expected intermediate variables, that is the effector position, with the help of the unparametrized model. Figure 8a shows, in green, various ISV solutions given by the student generator, after training. The best solution, with respect to the discriminator, is also fully displayed. As a baseline, solutions from an untrained student are also displayed in white. As can be seen, the trained solution distribution is more

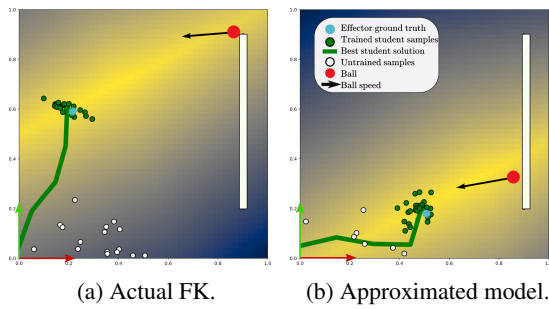


Figure 8: Sampled solutions for actual and approximated ISV in green. Baseline from untrained generator in white.

compact and centered in the area most valued by the teacher discriminator, as opposed to the untrained distribution. This demonstrates that it is possible to guide a generator with a discriminator through intermediate variables. Furthermore, it is possible to notice that the trained generator solution results in a student effector close to the original expert solution (light blue).

5.2 Approximated Kinematics ISV

Focus is now set on the **Approximated Kinematics** configuration. In this case, an approximation model is trained to predict the student effector position given the agent joint angles. Using the same teacher discriminator model as the **Actual Kinematics** stage, the student generator is trained similarly, replacing this time the agent analytical kinematic model by the trained one. Similar to the previous case, Figure 8b shows trained generator solutions and their baseline. As can be observed, the new generator solution distribution is also compact and adequately follows the discriminator preferences, demonstrating the reliability of the approximated model. While the approximated model is simple, this approach demonstrates that the CoachGAN is not limited to analytically defined models and can be used for cases where no robot kinematic model is available and must then be learned.

5.3 Ball Rebound Speed ISV

Finally, let us consider the last configuration where instead of relying on the effector position, the discriminator uses the **Ball Rebound Speed** to distinguish between expert trajectories and synthetic/not suitable ones. Figure 9 displays the discriminator evaluation of effector positions based on the predictor outputs as introduced in Section 4. Due to the fact that teacher discriminator training involves the predictor, the trained discriminator preferences do not overlap the distributions observed in Figure 7. However,

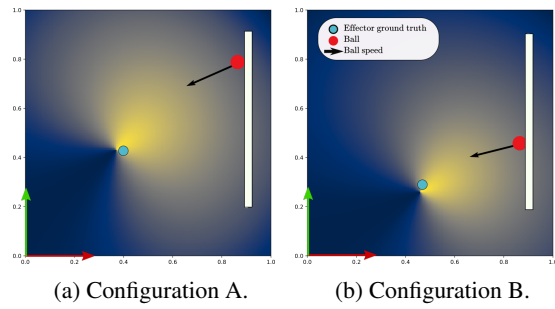


Figure 9: Task-Space evolution by the discriminator based on predictor results.

they still strongly favor positions located closely to the expert ground truth. The student generator is then trained accordingly to the process depicted in Figure 4

5.4 Analysis

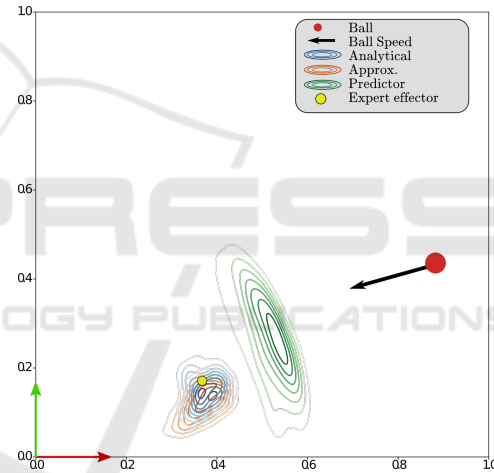


Figure 10: 2D representation of the solution distribution. Successful solutions are either overlapping expert position or in the ball path.

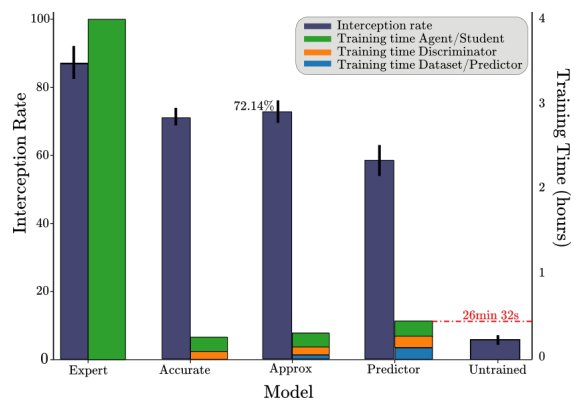


Figure 11: Interception performance and training time.

The performances of these models are displayed in Figure 10 where, for a given ball configuration, each model samples 2048 solutions. It is possible to notice that the blue and orange distribution, corresponding respectively to the analytical model and the approximated kinematics model, are very close, showing the approximated model reliability for transferring the discriminator evaluation. Moreover, these distributions overlap the expert effector position. The green distribution, representing solutions from the generator trained with the predictor pipeline is slightly broader than the two other distributions, as well as being located closer to the ball. As training for this model relied on another dataset, it consequently generated a differently inclined discriminator, leading in turn to a closer player student. Still, it appears that these distributions are directly located in the ball speed direction, thus enhancing the agent chances of intercepting the ball. Furthermore, it is possible to explain the width of the distributions by the fact that only the effector position is registered on contact and that the model is not directly aware of the bat length. Nonetheless, as shows Figure 11, all generators provide educated solutions. To establish these results, 500 ball trajectories are recorded, and, for every case, each student samples one joint angles solution given the initial ball configuration. Then, a check for collision is run along the ball path allowing *in fine* to obtain the interception score of each model (dark blue bar in Figure 11).

Now, let us consider the robots involved in the transfer experiments detailed above. While it might appear that these agents are kinematically close, it is important to note that the ISV vectors could have been generated by any robot having an end effector. Indeed, the student generator never has access to the expert geometry, thus broadening the spectrum of transfer.

Lastly, training these models requires negligible time and computation resources. Indeed, while the initial expert training requires 4 hours, the most time-consuming configuration, featuring additional dataset and predictor needs less than 12% of total training time (26.51 minutes). Student training in itself can be completed in less than ten minutes. And, as was shown in the **Actual Kinematics** case, a discriminator can be reused, thus further reducing the time needed for transfer. These results clearly illustrate that the CoachGAN method does allow fast transfer of task knowledge between differently shaped entities.

6 CONCLUSION

In this work, a new step-by-step methodology for systematical and fast transfer of knowledge from one entity to a differently structured one is proposed. Relying on an adversarial framework, the CoachGAN technique was evaluated using various intermediate state variables (ISV) and shows that it is possible to transfer task knowledge to a student agent without interaction of this agent with the task. As explained in the experiment results, this method allows training student within minutes, while most RL tasks require several hours of training. Thus, the applications of CoachGAN are numerous, especially given the increasing availability of approximation models. While many research directions are appealing, future works will focus on the integration of agent-related constraints within the loss function to progressively bridge the gap between simulation and reality.

ACKNOWLEDGEMENTS

This work has been sponsored by the French government research program Investissements d’Avenir through the RobotEx Equipment of Excellence (ANR-10-EQPX-44) and the IMobs3 Laboratory of Excellence (ANR-10-LABX-16-01), by the European Union through the program of Regional competitiveness and employment 2007-2013 (ERDF - Auvergne Region), by the Auvergne region and the French Institute for Advanced Mechanics.

REFERENCES

- Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., and Mordatch, I. (2019). Emergent Tool Use From Multi-Agent Autocurricula. *arXiv e-prints*, page arXiv:1909.07528.
- Boyden, S. (2004). *The Biology of Civilisation: Understanding Human Culture as a Force in Nature*. A UNSW Press book. University of New South Wales Press.
- Cruz, F., Parisi, G. I., Twiefel, J., and Wermter, S. (2016). Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Daejeon, South Korea, October 9-14*, pages 759–766. IEEE.
- Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

- Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. *CoRR*, abs/1802.06070.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. *ArXiv e-prints*.
- Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *ArXiv e-prints*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv e-prints*, page arXiv:1512.03385.
- Jeong, H. and Lee, D. D. (2016). Efficient learning of stand-up motion for humanoid robots with bilateral symmetry. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Daejeon, South Korea, October 9-14*, pages 1544–1549. IEEE.
- Martínez-González, Á., Villamizar, M., Canévet, O., and Odobez, J. (2018). Real-time convolutional networks for depth-based human pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Madrid, Spain, October 1-5*, pages 41–47. IEEE.
- Mounsif, M., Lengagne, S., Thuilot, B., and Adouane, L. (2019). Universal Notice Network: Transferable Knowledge Among Agents. *6th 2019 International Conference on Control, Decision and Information Technologies (IEEE-CoDIT)*.
- OpenAI, Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., Schneider, J., Tezak, N., Tworek, J., Welinder, P., Weng, L., Yuan, Q., Zaremba, W., and Zhang, L. (2019). Solving Rubik’s Cube with a Robot Hand. *arXiv e-prints*, page arXiv:1910.07113.
- Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016a). Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16*, page 2234–2242, Red Hook, NY, USA. Curran Associates Inc.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., and Chen, X. (2016b). Improved techniques for training gans. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.
- Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., van den Driessche, G., Schrittwiser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2015). Mastering the game of go with deep neural networks and tree search. *The Journal of Nature*.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.
- Trapit, B., Jakub, P., Szymon, S., Ilya, S., and Igor, M. (2017). Emergent complexity via multi-agent competition. *arXiv - OpenAI Technical Report*.
- Zintgraf, L. M., Shiarlis, K., Kurin, V., Hofmann, K., and Whiteson, S. (2018). CAML: Fast Context Adaptation via Meta-Learning. *ArXiv e-prints*.